Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HC08
Core Size	8-Bit
Speed	6MHz
Connectivity	I²C, USB
Peripherals	POR, PWM
Number of I/O	32
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 6x8b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68hc908bd48ib

Section 5. Configuration Register (CONFIG)

Section 6. Central Processor Unit (CPU)

Section 14. Multi-Master IIC Interface (MMIIC)

Section 15. DDC12AB Interface

Section 19. Computer Operating Properly (COP)

Section 20. Break Module (BRK)

- System protection features:
  - Optional computer operating properly (COP) reset
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- FLASH memory security[1]
- Master reset pin with internal pull-up and power-on reset
- $\overline{\text{IRQ}}$ with programmable pull-up and schmitt-trigger input
- 42-pin SDIP and 44-pin QFP packages

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (bcd) instructions
- Optimization for controller applications
- Third party C language support

1.4 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908BD48.

1. No security feature is absolutely secure. However, Freescale Freescale ’s strategy is to make reading or copying the FLASH difficult for unauthorized users.

5 are set permanently to logic 1. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	V	1	1	H	I	N	Z	C
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

Figure 6-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

Table 6-1. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,SP	Compare A with M	(A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,SP	Compare X with M	(X) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr,*X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7

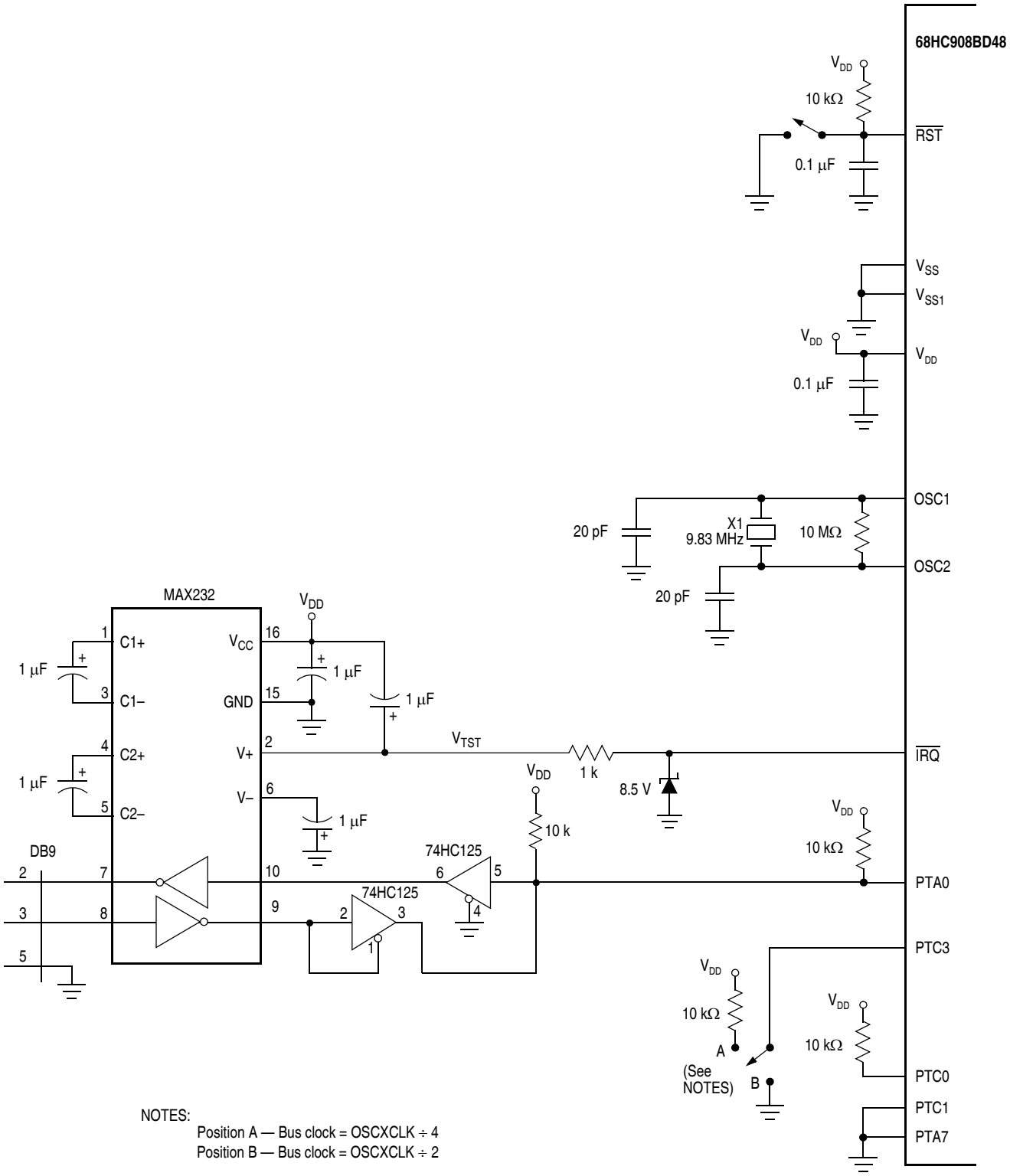
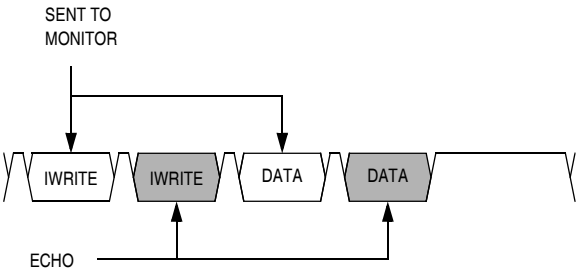


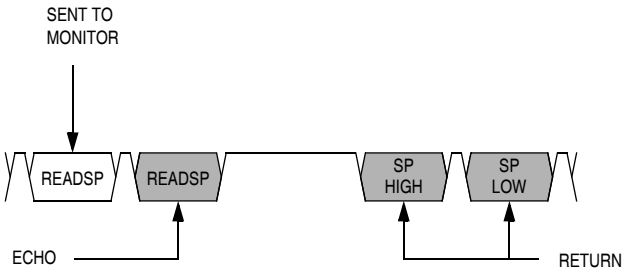
Figure 9-1. Monitor Mode Circuit

Table 9-6. IWRITE (Indexed Write) Command

Description	Write to last address accessed + 1
Operand	Specifies single data byte
Data Returned	None
Opcode	$19
Command Sequence	

A sequence of IREAD or IWRITE commands can sequentially access a block of memory over the full 64-kbyte memory map.

Table 9-7. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns stack pointer in high byte:low byte order
Opcode	$0C
Command Sequence	

Section 10. Timer Interface Module (TIM)

10.1 Contents

clearing sequence is complete, then writing logic zero to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a logic one to TOF has no effect.

1 = TIM counter has reached modulo value
0 = TIM counter has not reached modulo value

TOIE — TIM Overflow Interrupt Enable Bit

This read/write bit enables TIM overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIM overflow interrupts enabled
0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit

This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.

1 = TIM counter stopped
0 = TIM counter active

NOTE: *Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode.*

TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. Setting TRST has no effect on any other registers. Counting resumes from $0000. TRST is cleared automatically after the TIM counter is reset and always reads as logic zero. Reset clears the TRST bit.

1 = Prescaler and TIM counter cleared
0 = No effect

NOTE: *Setting the TSTOP and TRST bits simultaneously stops the TIM counter at a value of $0000.*

PS[2:0] — Prescaler Select Bits

These read/write bits select either the TCLK pin or one of the seven prescaler outputs as the input to the TIM counter as **Table 10-3** shows. Reset clears the PS[2:0] bits.

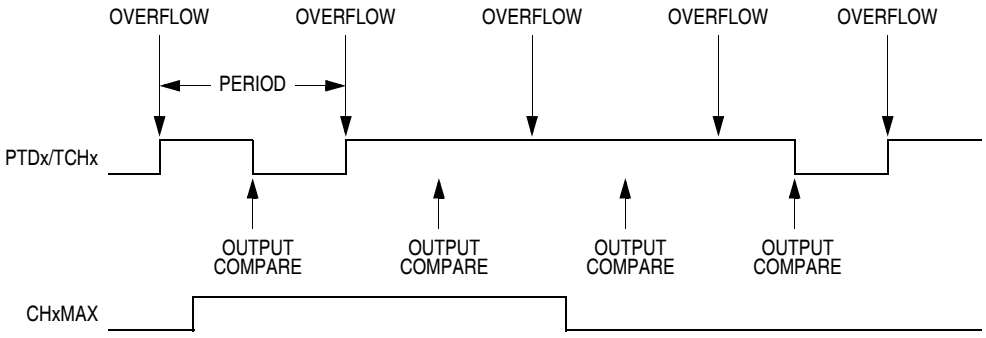


Figure 10-7. CHxMAX Latency

10.10.5 TIM Channel Registers (TCH0H/L:TCH1H/L)

These read/write registers contain the captured TIM counter value of the input capture function or the output compare value of the output compare function. The state of the TIM channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIM channel x registers (TCHxH) inhibits input captures until the low byte (TCHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIM channel x registers (TCHxH) inhibits output compares until the low byte (TCHxL) is written.

Combining the 5-bit PWM together with the 3-bit BRM, the average duty cycle at the output will be (M+N/8)/32, where M is the content of the 5-bit PWM portion, and N is the content of the 3-bit BRM portion. Using this mechanism, a true 8-bit resolution PWM type DAC with reasonably high repetition rate can be obtained.

The value of each PWM Data Register is continuously compared with the content of an internal counter to determine the state of each PWM channel output pin. Double buffering is not used in this PWM design.

Table 11-1. PWM I/O Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	PWM0 Data Register (0PWM)	Read: Write:	0PWM4	0PWM3	0PWM2	0PWM1	0PWM0	0BRM2	0BRM1	0BRM0
$0021	PWM1 Data Register (1PWM)	Read: Write:	1PWM4	1PWM3	1PWM2	1PWM1	1PWM0	1BRM2	1BRM1	1BRM0
$0022	PWM2 Data Register (2PWM)	Read: Write:	2PWM4	2PWM3	2PWM2	2PWM1	2PWM0	2BRM2	2BRM1	2BRM0
$0023	PWM3 Data Register (3PWM)	Read: Write:	3PWM4	3PWM3	3PWM2	3PWM1	3PWM0	3BRM2	3BRM1	3BRM0
$0024	PWM4 Data Register (4PWM)	Read: Write:	4PWM4	4PWM3	4PWM2	4PWM1	4PWM0	4BRM2	4BRM1	4BRM0
$0025	PWM5 Data Register (5PWM)	Read: Write:	5PWM4	5PWM3	5PWM2	5PWM1	5PWM0	5BRM2	5BRM1	5BRM0
$0026	PWM6 Data Register (6PWM)	Read: Write:	6PWM4	6PWM3	6PWM2	6PWM1	6PWM0	6BRM2	6BRM1	6BRM0
$0027	PWM7 Data Register (7PWM)	Read: Write:	7PWM4	7PWM3	7PWM2	7PWM1	7PWM0	7BRM2	7BRM1	7BRM0
$0028	PWM Control Register 1 (PWMCR1)	Read: Write:	PWM7E	PWM6E	PWM5E	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E
		Reset:	0	0	0	0	0	0	0	0

Table 11-1. PWM I/O Register Summary

$0051	PWM8 Data Register (8PWM)	Read:	8PWM4	8PWM3	8PWM2	8PWM1	8PWM0	8BRM2	8BRM1	8BRM0
		Write:								
$0052	PWM9 Data Register (9PWM)	Read:	9PWM4	9PWM3	9PWM2	9PWM1	9PWM0	9BRM2	9BRM1	9BRM0
		Write:								
$0053	PWM10 Data Register (10PWM)	Read:	10PWM4	10PWM3	10PWM2	10PWM1	10PWM0	10BRM2	10BRM1	10BRM0
		Write:								
$0054	PWM11 Data Register (11PWM)	Read:	11PWM4	11PWM3	11PWM2	11PWM1	11PWM0	11BRM2	11BRM1	11BRM0
		Write:								
$0055	PWM12 Data Register (12PWM)	Read:	12PWM4	12PWM3	12PWM2	12PWM1	12PWM0	12BRM2	12BRM1	12BRM0
		Write:								
$0056	PWM13 Data Register (13PWM)	Read:	13PWM4	13PWM3	13PWM2	13PWM1	13PWM0	13BRM2	13BRM1	13BRM0
		Write:								
$0057	PWM14 Data Register (14PWM)	Read:	14PWM4	PWM3	14PWM2	14PWM1	14PWM0	14BRM2	14BRM1	14BRM0
		Write:								
$0058	PWM15 Data Register (15PWM)	Read:	15PWM4	15PWM3	15PWM2	15PWM1	15PWM0	15BRM2	15BRM1	15BRM0
		Write:								
$0059	PWM Control Register 2 (PWMCR2)	Read:	PWM15E	PWM14E	PWM13E	PWM12E	PWM11E	PWM10E	PWM9E	PWM8E
		Write:								
		Reset:	0	0	0	0	0	0	0	0

11.4 PWM Registers

The PWM module uses of 18 registers for data and control functions.

- 16 PWM data registers ($0020–$0027 and $0051–$0058)
- 2 PWM control registers ($0028 and $0059)

TBIE — Transmit Buffer Interrupt Enable

When this bit is set, the TBEF flag is enabled to generate an interrupt request to the CPU. When TBIE is cleared, the TBEF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = TBEF bit set will generate interrupt request to CPU
0 = TBEF bit set does not generate interrupt request to CPU

RBIE — Receive Buffer Interrupt Enable

When this bit is set, the RBFF flag is enabled to generate an interrupt request to the CPU. When RBIE is cleared, the RBFF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = RBFF bit set will generate interrupt request to CPU
0 = RBFF bit set does not generate interrupt request to CPU

EOPIE — End Of Packet Interrupt Enable

When this bit is set, the EOPIF flag is enabled to generate an interrupt request to the CPU. When EOPIE is cleared, the EOPIF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = EOPIF bit set will generate interrupt request to CPU
0 = EOPIF bit set does not generate interrupt request to CPU

RSTIE —þReset Interrupt Enable

When this bit is set, the RSTIF flag is enabled to generate an interrupt request to the CPU. When RSTIE is cleared, the RSTIF flag is prevented from generating an interrupt request to the CPU. Reset clears this bit.

1 = RSTIF bit set will generate interrupt request to CPU
0 = RSTIF bit set does not generate interrupt request to CPU

NOTE: *Since there are more than one interrupt flags in the register, it is possible that program use Read-Modify-Write instruction to clear one flag, will occasionally clear the other flags which was just set after Read cycle of Read-Modify-Write operation.*

RSEQ — Received Data Sequence

This bit indicates the type of data packet of the last received data packet on endpoint 0. RSEQ = 0 indicates the last received data packet is type DATA0. RSEQ = 1 indicates the last received data packet is type DATA1.

1 = Last token received on endpoint 0 is a DATA1 token
0 = Last token received on endpoint 0 is a DATA0 token

SETUP — SETUP Token

This bit is set when the received token packet for endpoint 0 is a SETUP token (PID = b1101).

1 = Last token received on endpoint 0 is a SETUP token
0 = Last token received on endpoint 0 is not a SETUP token

TX1ST — Transmit First Flag

This bit is set if the endpoint 0 transmit buffer empty flag (TBEF) is set when the control logic is setting the endpoint 0 receive buffer full flag (RBFF). This happens when TBEF is still set at the end of an endpoint 0 reception.

1 = IN transaction occurred before SETUP or OUT
0 = IN transaction occurred after SETUP or OUT

RPSIZ[3:0] — Received Data Size

The RPSIZ[3:0] indicates the number of received data bytes in a data packet. Reset will not affect these bits

If the calling master does not return an acknowledge bit (MMRXAK = 1), the module will release the SDA line for master to generate a "stop" or "repeated start" condition. The data in the MMDTR will not be transferred to the output circuit until the next calling from a master. The transmit buffer empty flag remains cleared (MMTXBE = 0).

In master mode, the data in MMDTR will be transferred to the output circuit when:

- the module receives an acknowledge bit (MMRXAK = 0), after setting master transmit mode (MMRW = 0), and the calling address has been transmitted; or
- the previous data in the output circuit has be transmitted and the receiving slave returns an acknowledge bit, indicated by a received acknowledge bit (MMRXAK = 0).

If the slave does not return an acknowledge bit (MMRXAK = 1), the master will generate a "stop" or "repeated start" condition. The data in the MMDTR will not be transferred to the output circuit. The transmit buffer empty flag remains cleared (MMTXBE = 0).

The sequence of events for slave transmit and master transmit are illustrated in **Figure 14-7**.

14.5.6 Multi-Master IIC Data Receive Register (MMDRR)

Address: $004F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	MMRD7	MMRD6	MMRD5	MMRD4	MMRD3	MMRD2	MMRD1	MMRD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-6. Multi-Master IIC Data Receive Register (MMDRR)

When the MMIIC module is enabled, MMEN = 1, data in this read-only register depends on whether module is in master or slave mode.

In slave mode, the data in MMDRR is:

NOTE: *Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.*

Table 17-1. I/O Port Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA)	Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB)	Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC)	Read:	0	0	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD)	Read:	0	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA)	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB)	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC)	Read:	0	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

17.3 Port A

Port A is an 8-bit special-function port that shares all eight of its pins with the pulse width modulator (PWM).

17.3.1 Port A Data Register

The port A data register (PTA) contains a data latch for each of the eight port A pins.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by reset							
Alternate Function:	PWM15	PWM14	PWM13	PWM12	PWM11	PWM10	PWM9	PWM8

Figure 17-1. Port A Data Register (PTA)

PTA7–PTA0 — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

PWM15–PWM8 — PWM Outputs 15–8

The PWM output enable bits PWM15E–PWM8E, in PWM control register 2 (PWMCR2) enable port A pins as PWM output pins. (See **17.3.3 Port A Options**.)

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = (When read) Break address match

0 = (When read) No break address match

20.6.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: $FE0C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 20-3. Break Address Register High (BRKH)

Address: $FE0D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Figure 20-4. Break Address Register Low (BRKL)

20.6.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

21.4 Functional Operating Range

Characteristic	Symbol	Value	Unit
Operating Temperature Range	T_A	0 to 85	°C
Operating Voltage Range	V_{DD}	4.5 to 5.5	V

21.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance QFP (44 Pins) SDIP (42 Pins)	θ_{JA}	 95 60	°C/W
I/O Pin Power Dissipation	$P_{I/O}$	User Determined	W
Power Dissipation(1)	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} = K/(T_J + 273\ °C)$	W
Constant(2)	K	$P_D \times (T_A + 273\ °C) + P_D^2 \times \theta_{JA}$	W/°C
Average Junction Temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C
Maximum Junction Temperature	T_{JM}	100	°C

NOTES:

1. Power dissipation is a function of temperature.
2. K is a constant unique to the device. K can be determined for a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.